IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Serial No. 09/740,531
Confirmation No. 8383

I hereby certify that this correspondence is being transmitted to the United States Patent & Trademark Office via electronic submission or facsimile on the date indicated below:

01/08/2007 /Pamela Gerik/
Date Pamela Gerik

SUPPLEMENTAL APPEAL BRIEF

Sir/Madam:

Further to the Notice of Appeal faxed January 21, 2005 and received in the U.S. Patent and Trademark Office on the same day, Appellant presents this Supplemental Appeal Brief in response to a Notice of Non-Compliant Brief mailed December 14, 2006. The Notice of Appeal was filed following mailing of a final Office Action on October 22, 2004. Appellant hereby appeals to the Board of Patent Appeals and Interferences from a final rejection of pending claims 1-17 and 19-21, and respectfully requests that this appeal be considered by the Board.

I. REAL PARTY IN INTEREST

The subject application is owned by International Business Machines Corporation, a corporation having its principal place of business at New Orchard Road, Armonk, New York, 10504, as evidenced by the assignment recorded at Reel 011429, Frame 0838.

II. RELATED APPEALS AND INTERFERENCES

A Notice of Appeal has been filed for the following application, which shares a common specification with the application currently on appeal.

09/740,460 Notice of Appeal filed on or about January 18, 2005.

However, because dissimilar art is cited in the present application and the above-mentioned related application, Appellants do not believe that the outcome of this appeal will have any bearing on the Board's decision on the related appeal. No other appeals or interferences are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-21 were originally filed. Claim 18 was canceled. Claims 1-17 and 19-21 are pending and stand rejected.

IV. STATUS OF AMENDMENTS

No amendments to the claims have been filed subsequent to their final rejection. The Appendix hereto therefore reflects the current state of the claims.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Appellant's claimed invention relates to a software system (claim 1), computer program product (claim 19), a server (claim 21) and a method (claim 10) for detecting affinity breaks between a client and a server. An "affinity condition" exists when a client's requests are all routed to the same server. An "affinity break" occurs when a server designated for processing the client's requests goes down for some reason. When affinity is broken with the original server, the user's requests may be temporarily redirected to a different server. When the original

server is operational again, affinity may be restored (Specification -- page 15, lines 4-15; page 24, line 37 – page 25, line 16).

In some embodiments, the software system as recited in present claim 1 may be configured for supporting client/server affinity detection. For example, the software system may include a server (e.g., web server 14a, Fig. 1) and a client (e.g., client 20a, Fig. 1), which is adapted to send requests to the server (Specification -- page 5, line 29 – page 6, line 34). In addition, the software system may include numeric-valued "generation ID" (or GID) with each request sent from the client to the server (Specification -- page 15, lines 16-21; page 25, lines 17-21). The GID is incremented by the server upon receiving the request, and recorded by the server before being returned to the client (Specification -- page 15, lines 21-28; page 25, lines 21-34). Thus, if the GID accompanying a request from the client differs from the GID recorded by the server, an affinity break between the client and the server is detected (Specification -- page 15, lines 28-35; page 25, line 34 – page 26, line 7).

In some embodiments of the invention, the method as recited in claim 10 may detect affinity breaks between a client (e.g., client 20a, Fig. 1) and a server (e.g., web server 14a, Fig. 1) (Specification -- page 5, line 29 – page 6, line 34). For example, the method may include the client sending a request to the server, where the request is accompanied by a numeric-valued generation ID (GID) (Specification -- page 15, lines 16-21; page 25, lines 17-21). In addition, the method may include the server receiving the request and the GID from the client, and comparing the received GID against a previously recorded GID. If the received GID matches the recorded GID, the server may increment the recorded GID and return the incremented GID to the client as a new GID in a third step of the method (Specification -- page 15, lines 21-28; page 25, lines 21-34). However, if the received GID does not match the recorded GID, an affinity break may be reported between the client and the server in a fourth step of the method. (Specification -- page 15, lines 28-35; page 25, line 34 – page 26, line 7).

In some embodiments of the invention, the computer program product as recited in claim 19 may be included within a computer readable medium (e.g., a RAM, ROM or hard disk drive) for use in detecting affinity breaks between a client (e.g., client 20a, Fig. 1) and a server (e.g.,

web server 14a, Fig. 1) (Specification -- page 5, line 29 – page 6, line 34). For example, the computer program product may include instructions for the server receiving a request and a numeric-valued generation ID (GID) from the client, and comparing the received GID against a previously recorded GID (Specification -- page 15, lines 16-24). In addition, the computer program product may include instructions for incrementing the recorded GID, and returning it to the client as the new GID, if the received GID matches the recorded GID (Specification -- page 15, lines 24-28). Furthermore, the computer program product may include instructions for reporting an affinity break between the client and the server, if the received GID does not match the recorded GID (Specification -- page 15, lines 28-35).

In some embodiments of the invention, the server as recited in claim 21 may include memory and a processor for detecting affinity breaks (Specification -- page 5, line 29 – page 6, line 34). For example, the server (e.g., web server 14a, Fig. 1) may include means for receiving a request and a numeric-valued generation ID (GID) from a client (e.g., client 20a, Fig. 1), and for comparing the received GID against a previously recorded GID (Specification -- page 15, lines 16-24). In some cases, the server may include means for incrementing the recorded GID, and returning it to the client as the new GID, if the received GID matches the recorded GID (Specification -- page 15, lines 24-28). In some cases, the server may include means for reporting an affinity break between the client and the server, if the received GID does not match the recorded GID (Specification -- page 15, lines 28-35).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-3, 5-7, 10-12, 14-16, and 19-21 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 5,706,435 to Barbara et al. (hereinafter "Barbara").

2. Claims 4, 8, 9, 13, and 17 stand rejected under 35 U.S.C § 103(a) as being unpatentable over Barbara in view of U.S. Patent No. 6,327,628 to Anuff et al. (hereinafter "Anuff").

VII. ARGUMENT

The contentions of the Appellant with respect to the ground of rejection presented for review, and the basis thereof, with citations of the statutes, regulations, authorities, and parts of the record relied upon are presented herein for consideration by the Board. Details as to why the rejections cannot be sustained are set forth below.

A. Patentability of claims 1-3, 5-7, 10-12, 14-16 and 19-21 under 35 U.S.C § 102(e)

Claims 1-3, 5-7, 10-12, 14-16, and 18-21 were rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 5,706,435 to Barbara et al. (hereinafter "Barbara"). Claim 18 was canceled in a previous response rendering rejection thereto moot. As described in more detail below, the § 102(e) rejection of claims 1-3, 5-7, 10-12, 14-16, and 19-21 is hereby traversed. The standard for "anticipation" is one of fairly strict identity. A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art of reference. *Verdegaal Bros. v. Union Oil Co. of California*, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987); MPEP 2131. Using this standard, Applicants submit the cited art fails to disclose each and every element of the currently pending claims, some distinctive features of which are set forth in more detail below.

Barbara fails to disclose a client that sends a request to a server, where the request includes a numeric-valued generation ID (GID), as recited in claims 1, 10, 19, and 21. Independent claims 1, 10, 19, and 21 each set forth a communication path from a client to a server. Specifically, the client is claimed to send a "request" to the server. Included with that request is a numeric-valued generation ID, or GID. The Specification describes the GID as a unique value, which is assigned to each request sent from the client to the server to represent the current state of the request (*see*, Specification; page 15, lines 16-21; page 25, lines 20-26). The Final Office Action alleges that the item labeled "information identifying" in Barbara is the same as the presently claimed GID (*see*, e.g., Final Office Action, pages 2 and 4). Appellants disagree.

Statements in the final Office Action suggest that

> Barbara et al. In COL 2, LINES 60-65 discloses that the server processor periodically broadcasts invalidation reports to the client processor. Each respective invalidation report includes information identifying which, if any, of the plurality of data values have been updated within a predetermined period of time before the server processor broadcasts the respective invalidation report [to the client] (Final Office Action, page 2).

Further statements in the final Office Action suggest that "[i]t is very well known that an ID is "identifying information", which could come in the form of a numeric-value" (Final Office Action, page 2). As described in more detail below, the Appellants strongly disagree with the Examiner's contention that the "invalidation reports" of Barbara, or the "information" contained therein, can be considered equivalent to the presently claimed requests or numeric-valued GID.

First of all, Barbara clearly and repeatedly states that the "invalidation report" (and thus, the information contained therein) is sent or "broadcast" from a server processor to a client processor (*see*, e.g., Barbara, Abstract; col. 2, lines 60-65; col. 4, lines 5-26; col. 6, lines 5-40; col. 7, line 66 – col. 8, line 26; col. 10, lines 1-22). The Examiner notes such a fact on page 2 of the Office Action mailed April 9, 2004 and on page 2 of the final Office Action mailed October 22, 2004. Appellants assert that since the so-called "identifying information" contained within the "invalidation report" of Barbara is sent from a server processor – and not from a client, as presently claimed – the "identifying information" of Barbara cannot be included within a request sent from a client, and therefore, cannot be the same as or equivalent to the presently claimed GID, which is sent in a request from a client to a server.

Second, Barbara fails to disclose that the invalidation reports may contain a numeric-valued generation ID, which is described in the present Specification as a unique value associated with each request sent from a client to a server to represent the current state of the request (Specification -- page 15, lines 16-21; page 25, lines 20-26). As noted above, the Examiner suggests that "an ID is 'identifying information', which *could* come in the form of a numeric-value" (Final Office Action, page 2, emphasis added). Although this may be true, the Appellant disagrees with the liberties taken by the Examiner. Instead of the Examiner simply

making guesses based on hindsight, the burden is on the Examiner to find some suggestion for the claimed limitation in the teachings of the prior art reference. Barbara fails to provide teaching or suggestion for a request having an ID associated therewith, and provides even less teaching for the presently claimed numeric-valued generation ID.

Instead, Barbara discloses, "[e]ach invalidation report includes information identifying which of the data values stored in the server 10a-10d have been updated within a predetermined interval of time" (Barbara -- col. 4, lines 11-14). In various embodiments of the method, Barbara discloses that the "information identifying" may include: (i) a list of addresses corresponding to data updated in the server (Barbara -- col. 6, lines 5-25), (ii) a list of addresses and time stamps for the updated data in the server (Barbara -- col. 7, line 61 –o col. 8, line 17), or (iii) a list of combined "signatures" or checksums computed over the values of data in the server (Barbara -- col. 10, lines 1-22).

As such, Barbara makes clear that the "information" included within the invalidation reports may contain a list of addresses, a list of addresses and timestamps, or a list of combined signatures. Appellants contend that a list of any sort cannot be the same as a numeric-valued generation ID, which is described in the specification as a unique value assigned to each request sent from a client to a server. In at least one embodiment of the method, Barbara specifically states that the "report only includes the list of addresses" (Barbara -- col. 6, line 21, emphasis added). Therefore, even if the teachings of Barbara were modified to send invalidation reports from a client to a server (without sufficient motivation to do so), the invalidation reports of Barbara would still fail to include a numeric-valued generation ID (which is unique to each request). As such, the invalidation reports of Barbara cannot be considered equivalent to the presently claimed requests, nor can they be considered to include the presently claimed numeric-valued generation ID.

Barbara fails to disclose a server, which is adapted to receive the request from the client and to compare the received GID against a previously recorded GID, as recited in claims 10, 19 and 21. Each of independent claims 10, 19, and 21 teach that upon receiving the request from the client, the server may compare the received GID against a GID previously

recorded within the server to detect if an affinity break has occurred. The final Office Action alleges that the third embodiment of Barbara, referred to as "Broadcasting Signatures," provides teaching for "a server that compares the GID received from the client against a GID that was previously recorded in the server to detect if an affinity break has occurred" (Final Office Action, page 3). The Appellant disagrees.

First of all, and as noted above, Barbara discloses a system and method in which a client receives an invalidation report (an alleged "request") from a server. However, Barbara fails to disclose the presently claimed system or method in which a server receives a request from a client, where the request includes a numeric-valued generation ID (GID), which is unique to each request and sent along with each request transmitted between a client and server.

In addition to failing to receive a request and numeric-valued GID from a client, Barbara fails to disclose a server which functions to compare a received GID against a GID previously recorded in the server. The Examiner suggests that "Barbara on col 10, lines 1-7, discloses that 'Broadcasting Signatures' involves a comparison between … a first set of signatures based on the contents of the server 10a and a second set of signatures based on the contents of the cache 22 of client 20a" (Final Office Action, page 3). First of all, the "signatures" disclosed by Barbara are not numeric-valued generation IDs, or GIDs. Therefore, the signature comparison disclosed by Barbara cannot be equivalent to the presently claimed system and method for comparing a received GID with a previously recorded GID. In addition, Barbara clearly states that the signature comparison is performed by the client, and not by a server, as presently claimed. Statements in the final Office Action even point out this fact by stating, "Barbara… discloses that at step 286, the client 20a compares its combined signature to the combined signature received in the invalidation report" (Final Office Action, page 3). For at least these reasons, the server processor of Barbara cannot be considered equivalent to the presently claimed server, as recited in claims 10, 19, and 21.

Barbara fails to disclose that, if the received GID matches the recorded GID, the server increments the recorded GID and returns the incremented GID to the client as a new GID, as recited in claims 1, 10, 19, and 21. Each of independent claims 1, 10, 19, and 21

teach that, if the GID received by the server (from the client) matches the GID previously recorded in the server, the server will increment the GID and return the incremented GID to the client as a new GID. This is done to ensure that the GID represents the current state of the request, and may be used by the server to determine if the server has missed any requests sent from the client (Specification -- page 15, lines 21-25). The final Office Action alleges that Barbara provides teaching for the present claim limitation by disclosing a method for comparing the combined signatures in the invalidation report with those stored in the cache memory of a client, and a counter which is incremented for each datum represented by the non-matching combined signatures (Final Office Action, page 2). The Appellant disagrees.

First of all, the method steps referred to by the Examiner (e.g., steps 286-290, as shown in Fig. 5B of Barbara) are only performed by the clients of Barbara (Barbara -- col. 10, lines 23-25), and not be a server, as presently claimed.

Second, Barbara fails to disclose a server, which is adapted to compare a GID received by the server (from a client) with a GID previously recorded in the server, and therefore, cannot provide teaching or suggestion for a server that performs a particular function, if the received GID matches the previously recorded GID.

Third, Barbara fails to disclose that a previously recorded GID (or any GID, for that matter) may be incremented by the server processor (or any other computational unit within the system of Barbara) to form a new GID. Instead, Barbara discloses (and the Examiner points out) that a counter may be incremented for each datum represented by the non-matching combined signatures (Barbara -- col. 11, lines 1-10; Final Office Action, page 2). Though Barbara discloses that a counter may be incremented (in step 290) to determine the "number of non-matching combined signatures associated with each respective datum" in the cache (Barbara -- col. 11, lines 10-15), neither the counter nor the value stored therein may be considered equivalent to the presently claimed "recorded GID." For example, Barbara fails to disclose that the counter value may be a unique value, which is transmitted along with each request sent from a client to a server.

Fourth, Barbara fails to disclose that the new GID (i.e., the GID formed by incrementing the recorded GID) may be returned to the client. Though Barbara discloses that a counter may be incremented (in step 290), the incremented count value stored therein is not an incremented GID and is not returned to the client as a new GID.

Barbara fails to disclose that an affinity break is detected/reported between the client and the server, if the received GID does not match the recorded GID, as recited in claims 1, 10, 19 and 21. Each of independent claims 1, 10, 19, and 21 teach that an affinity break may be detected (claim 1) or reported (claims 10, 19 and 21), if the GID received by the server (from the client) does not match the GID previously recorded in the server. The final Office Action alleges that the "Broadcasting Signatures" method shown in Figs. 5A and 5B of Barbara provides teaching for detecting or reporting an affinity break if a received GID does not match a recorded GID. The Appellants disagree.

Barbara discloses an invalidation process, which allows a client processor to invalidate select data values stored in the cache memory of the client, if the data values were updated in a server processor after the values were stored in the cache memory of the client (*see*, Barbara, Abstract). However, Barbara fails to disclose that an affinity break may be detected or reported between a client and a server, if a received GID (from the client) does not match a recorded GID (in the server). In fact, Barbara has absolutely nothing to do with detecting or reporting affinity breaks between a client and a server, regardless of the manner in which detection is performed.

In light of the Examiner's suggestions in the Final Office Action, it appears that the Examiner does not understand the difference between an "affinity break," as set forth in the present claims, and "invalidation" in general. Details of what would constitute an affinity break and the differences between client/server affinity and cache incoherency/invalidation are set forth throughout the present specification (Specification -- page 15, lines 4-35; page 24, line 37 – page 26, line 7). In general, affinity breaks are provided to determine whether communication between a particular client and a particular server has been interrupted. If communication has been interrupted, the client data stored in the cache memory of a server may be invalid. To determine if an affinity break has occurred, a GID can be used that is unique to each request sent

from the client to the server. The GID sent from the client can be compared with a stored GID to determine if a break has occurred. As each request is received, the server will increment its GID and a break is encountered if the received and stored GIDs (as incremented) do not match. Comparing GIDs and detecting affinity breaks occurs well before any determination is needed on which pieces of data stored in cache should be invalidated.

Barbara only deals with instances in which invalid data are known and a report of such data (i.e., an invalidation report) is present. This is entirely non-analogous to affinity breaks, or the problem solved by the present claims of determining a break in communication between a client and a server. Certainly, a skilled artisan would appreciate the difference between detecting communication breaks between a particular client and server (which are detected as affinity breaks) and sending invalidation reports to a client (possibly, after an affinity break has occurred) to allow the client to update data stored therein.

In other words, though the presently claimed method for detecting an affinity break may ultimately be used for invalidating contents (in the server), it is unreasonable for one skilled in the art to assume that, because Barbara discloses an invalidation process, the teachings of Barbara must inherently disclose a preceding process, which determines whether or not an affinity break has occurred between a client and a server. Barbara simply fails to disclose any process for detecting whether or not an affinity break has occurred, and therefore, cannot teach or suggest that an affinity break may be detected (or reported) if a GID received from a client does not match a GID previously recorded in the server, as presently claimed.

As noted above, Barbara fails to anticipate not just one, but substantially all limitations recited in independent claims 1, 10, 19 and 21. Therefore, Appellants assert that independent claims 1, 10, 19, 21, and all claims dependent therefrom, are not anticipated by the cited art. Appellants further assert that an anticipatory rejection of the current claims cannot be sustained and, therefore, request that this rejection be reversed.

B. Patentability of claims 4, 8, 9, 13, and 17 under 35 U.S.C § 103(a)

Claims 4, 8, 9, 13, and 17 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Barbara in view of U.S. Patent No. 6,327,628 to Anuff et al. (hereinafter "Anuff"). Because claims 4, 8, 9, 13 and 17 are dependent from independent claims 1 and 10, the arguments presented above for patentability of claims 1 and 10 apply equally to claims 4, 8, 9, 13 and 17, and are herein incorporated by reference. In addition to the 35 U.S.C. § 102 arguments presented above with respect to claims 1 and 10, arguments are provided below to establish patentability of the current claims under 35 U.S.C. § 103(a).

MPEP 2143 establishes the basic requirements in finding a *prima facie* case of obviousness. As described, to establish a case of *prima facie* obviousness of a claimed invention, three criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art to modify the references or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art references must teach or suggest all the claim limitations. Specifically, "all words in a claim must be considered when judging the patentability of that claim against the prior art." *In re Wilson* 424 F.2d. 1382, 1385 (CCPA 1970). If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988).

None of the cited art teaches or suggests a server adapted to: (i) receive a request including a numeric-valued generation ID (GID) from a client, (ii) compare the received GID against a GID previously recorded in the server, (iii) increment the recorded GID and return the incremented GID to the client as a new GID, if the received GID matches the recorded GID, or (iv) report an affinity break between the client and the server, if the received GID does not match the recorded GID, as recited in claims 1 and 10. For at least the reasons set forth above, Barbara fails to provide teaching or suggestion for a server, as recited in present claims 1 and 10. Even though Anuff is not cited against present claims 1 and 10, Appellants assert that the teachings of Anuff cannot be combined with those of Barbara to overcome the deficiencies therein.

Anuff discloses a portal server that presents an HTML page with a plurality of modules that are formatted in a predetermined layout (Anuff, Abstract). However, Anuff fails to disclose that the portal server (or any other server for that matter) may be adapted to: (i) receive a request including a numeric-valued generation ID (GID) from a client, (ii) compare the received GID against a GID previously recorded in the server, (iii) increment the recorded GID and return the incremented GID to the client as a new GID, if the received GID matches the recorded GID, or (iv) report an affinity break between the client and the server, if the received GID does not match the recorded GID. In other words, like Barbara, Anuff fails to disclose any of the limitations recited in claims 1 and 10.

Since Anuff fails to disclose the above-mentioned claim limitations, which are also absent from the teachings of Barbara, the teachings of Barbara and Anuff cannot be combined in a manner that would disclose all limitations recited in claims 1 and 10. In addition, Barbara and Anuff cannot be modified to teach or suggest the above-mentioned claim limitations, since neither Barbara nor Anuff provide teaching, suggestion or motivation to do so. Obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed.Cir. 1988); *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992); MPEP 2143.01.

None of the cited art provides teaching or suggestion for a server adapted to perform the functions, as recited in claims 1 and 10, and where the server comprises a Java Virtual Machine (JVM) equipped with a cache, as recited in claim 4. Claim 4, which includes all limitations recited in claim 1, places a further limitation on the presently claimed server to comprise a Java Virtual Machine (JVM) equipped with a cache.

Statements in the final Office Action admit that Barbara does "not explicitly disclose a Java Virtual Machine (JVM) equipped with a cache," but suggests that Anuff discloses "a method where a memory cache can be cleared by the Java Virtual Machine (JVM) when resources are running low" (Final Office Action, page 7). Therefore, the Examiner suggests that it would be "obvious to one of ordinary skills in the art at the time the invention was made to

incorporate Barbara's teaching of coherency in cache memory with [a] portal server using JVM [as] disclosed by Anuff because an improved method for maintaining a coherent view of the data in the cache of each mobile unit is desired" (Final Office Action, pages 7-8). As described below, the Appellants disagree that the combined teachings of Barbara and Anuff could somehow be interpreted to read upon the presently claimed server.

As noted above, Barbara and Anuff each fail to provide teaching or suggestion for a server that performs the functionality recited in claims 1 and 10. The Examiner admits that no teaching or suggestion can be found within Barbara for a server that includes a JVM equipped with a cache. Though Anuff may briefly mention that a "memory cache can be cleared by the Java Virtual Machine (JVM) when resources are running low" (Anuff -- col. 11, lines 61-63), the point is moot. Even if the portal server of Anuff (which may include a JVM) is combined with the teachings of Barbara (without sufficient motivation to do so), the combined teachings of Barbara and Anuff would still fail to disclose a server that performs the functionality recited in claims 1 and 10. Therefore, the combined teachings of Barbara and Anuff fail to disclose all limitations of dependent claim 4.

None of the cited art provides teaching or suggestion for a system (or method) in which a server is adapted to perform the functions, as recited in claims 1 and 10, and where the system comprises an object-oriented software system, as recited in claims 9 and 13. Claims 9 and 13, which include all limitations recited in claims 1 and 10, place a further limitation on the presently claimed system and method by reciting that the system comprises an object-oriented software system.

Statements in the final Office Action suggest that teaching for an object oriented software system may be found in column 4, lines 47-48 of Anuff (Final Office Action, page 8). Though Anuff does mention that an "object-oriented software system consists of software objects," Anuff does <u>not</u> disclose an object-oriented software system (or method), which includes a server adapted to perform the functions recited in claims 1 and 10. Since Barbara also fails to disclose such a server, the combined teachings of Barbara and Anuff cannot be relied upon to provide

teaching or suggestion for all limitations recited in claims 9 and 13, which include the limitations recited in base claims 1 and 10.

None of the cited art provides teaching or suggestion for an affinity command that may be sent by a server to a client (and returned by the client to the server) in the form of a cookie, as recited in claims 8 and 17. Claims 8 and 17 each recite an additional limitation that states an "affinity command is sent by the server to the client and returned by the client to the server in a cookie." Statements in the final Office Action suggest that the login information disclosed by Anuff "can be stored as a browser cookie so that users don't have to log in each time they visit a site" (Final Office Action, page 8). As such, the Examiner appears to suggest that the "login information" of Anuff is somehow equivalent to the presently claimed affinity command. The Appellants disagree.

In column 13, lines 25-31, Anuff discloses a "login page [that] enables users to identify themselves to the portal server by entering their user name and password. The login information can be stored as a browser cookie so that users don't have to log in each time they visit a site." Though Anuff mentions the use of a cookie, the login information stored in the cookie of Anuff (i.e., a user name and password) cannot be considered equivalent to an affinity command, which is described in the present claimed case as including a user ID and a generation ID, where the user ID is unique to each client and the generation ID is unique to each request sent from a client to a server (Specification -- page 15, lines 16-21; present claims 3 and 12;). Therefore, Appellants assert that Anuff fails to disclose the limitations recited in claims 8 and 17.

For the foregoing reasons, Appellant asserts that independent claims 1 and 10, as well as claims dependent therefrom, are patentably distinct over Barbara and Anuff. Contrary to the characterizations made in the various Office Actions, the cited references cannot be properly combined or modified to provide teaching for all limitations recited in claims 4, 8, 9, 13, and 17, especially since those claims include all limitations of base claims 1 and 10. Accordingly, Appellants assert that a *prima facie* case of obviousness has not been duly set out and, therefore, request that this rejection be reversed.

* * *

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-21 was erroneous, and reversal of the decision is respectfully requested.

The Commissioner is authorized to charge the required fees to Daffer McDaniel LLP deposit account number 50-3268/5468-05700.

Respectfully submitted,
/Kevin L. Daffer/
Kevin L. Daffer
Reg. No. 34,146
Attorney for Appellant

Customer No. 35617
Date: January 8, 2007
JMF

VIII. CLAIMS APPENDIX

The present claims on appeal are as follows.

1. A software system for distributed web applications, supporting client/server affinity detection, comprising:

a server;

a client, adapted to send requests to the server; and

a numeric-valued generation ID, accompanying each request from the client to the server, incremented by the server upon receiving the request, and recorded by the server before being returned to the client, and such that if the generation ID accompanying a request from the client differs from the generation ID recorded by the server, an affinity break between the client and the server is detected.

2. The software system as recited in claim 1, further comprising a plurality of clients adapted to send requests to the server, wherein each client has a unique user ID.

3. The software system as recited in claim 2, further comprising an affinity command, which combines the generation ID accompanying a request with the user ID of the client sending the request, and by means of which the server may detect an affinity break with a particular client among the plurality of clients.

4. The software system as recited in claim 3, wherein the server comprises a Java Virtual Machine (JVM) equipped with a cache.

5. The software system as recited in claim 4, further comprising a plurality of servers, wherein affinity between a client and first server may be broken as a result of the client sending a request to a second server.

6. The software system as recited in claim 4, wherein an affinity break between a client and a server may occur if the server becomes unavailable.

7. The software system as recited in claim 4, wherein detection of an affinity break between a client and a server may be used to invalidate contents of the cache in the server.

8. The software system as recited in claim 4, wherein the affinity command is sent by the server to the client and returned by the client to the server in a cookie.

9. The software system as recited in claim 1, further comprising an object-oriented software system.

10. A method for detecting affinity breaks between a client and a server equipped with a cache in a software system for distributed web applications, comprising:

the client sending a request to the server, accompanied by a numeric-valued generation ID (GID);

the server receiving the request and the GID from the client, and comparing the received GID against a previously recorded GID;

if the received GID matches the recorded GID, incrementing the recorded GID, and returning it to the client as the new GID; and

if the received GID does not match the recorded GID, reporting an affinity break between the client and the server.

11. The method as recited in claim 10, further comprising detecting affinity breaks between a plurality of clients and a server, wherein each client has a unique user ID.

12. The method as recited in claim 11, further comprising sending an affinity command with each request from a client, such that the affinity command combines the GID with the user ID of the client sending the request, and detecting an affinity break with a particular client among the plurality of clients by means of the user ID.

13. The method as recited in claim 11, wherein the software system comprises an object-oriented software system.

14. The method as recited in claim 12, further comprising detecting affinity breaks between a plurality of clients and a plurality of servers, each of which is equipped with a cache, such that affinity between a client and first server may be broken as a result of the client sending a request to a second server.

15. The method as recited in claim 14, wherein an affinity break between a client and a server may occur if the server becomes unavailable.

16. The method as recited in claim 15, wherein detection of an affinity break between a client and a server may be used to invalidate contents of the cache in the server.

17. The method as recited in claim 16, wherein the affinity command is sent by the server to the client and returned by the client to the server in a cookie.

19. A computer program product in a computer readable medium for use in detecting affinity breaks between a client and a server, the computer program product comprising:

instructions for the server receiving a request and a numeric-valued generation ID (GID) from the client, and comparing the received GID against a previously recorded GID;

instructions for incrementing the recorded GID, and returning it to the client as the new GID, if the received GID matches the recorded GID; and

instructions for reporting an affinity break between the client and the server, if the received GID does not match the recorded GID.

20. The product as recited in claim 19 further comprising:

instructions for the client sending a request to the server, accompanied by a numeric-valued generation ID (GID).

21. A server including memory and processor detecting affinity breaks, comprising:

means for the server receiving a request and a numeric-valued generation ID (GID) from the client;

comparing the received GID against a previously recorded GID;

means for incrementing the recorded GID, and returning it to the client as the new GID, if the received GID matches the recorded GID; and

means for reporting an affinity break between the client and the server, if the received GID does not match the recorded GID.

IX. EVIDENCE APPENDIX

No evidence has been entered during the prosecution of the captioned case.

X. RELATED PROCEEDINGS APPENDIX

A Notice of Appeal has been filed for the following application, which shares a common specification with the application currently on appeal.

09/740,460 Notice of Appeal filed on or about January 18, 2005.

However, because dissimilar art is cited in the present application and the above-mentioned related application, Appellants do not believe that the outcome of this appeal will have any bearing on the Board's decision on the related appeal. No other appeals or interferences are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.